

# ANÁLISIS Y EVALUACIÓN DE PLACAS DE DESARROLLO BASADAS EN RISC-V

Raúl Sierra Jiménez

**Resumen**— Dentro de la Universidad Autónoma de Barcelona se está empezando a realizar un impulso en la investigación y el desarrollo de hardware de código abierto basado en el conjunto de instrucciones RISC-V. Para colaborar con esta apuesta por el hardware libre dentro de la universidad, en este trabajo se ha realizado una investigación teórica sobre RISC-V conjuntamente con el análisis y evaluación de dos placas de desarrollo con la arquitectura de repertorio de instrucciones RISC-V para aplicar los conceptos adquiridos en el diseño y materialización de prácticas de laboratorio con la finalidad de introducir este ISA a los alumnos.

**Palabras clave**— RISC-V, hardware libre, repertorio de instrucciones, ensamblador, sistemas embebidos, microcontroladores, SiFive, SparkFun

**Abstract**— Inside the Universidad Autónoma de Barcelona, a boost is being made in the research and development of open-source hardware based on the RISC-V instruction set. To collaborate with this commitment to free hardware inside the university, in this paper a theoretical investigation has been carried out on RISC-V together with the analysis and evaluation of two development boards with the RISC-V instruction repertoire architecture to apply the concepts acquired in the design and materialization of laboratory practices in order to introduce this ISA to the students.

**Index Terms**— RISC-V, Open-source hardware, Instruction set architecture, assembler, embedded systems, microcontrollers, SiFive, SparkFun



## 1 INTRODUCCIÓN – CONTEXTO DEL TRABAJO

RISC-V es una arquitectura abierta y gratuita que nos ofrece la posibilidad de poder trabajar sin necesidad de pagar royalties en desde procesadores para microcontroladores de sistemas empujados hasta para sistemas de cómputo de altas prestaciones.

Existe una apuesta muy fuerte sobre esta arquitectura por parte de diferentes sectores, entre ellos la Unión Europea con el proyecto EPI [1] (European Processor Initiative) o universidades y centros de investigación a nivel nacional con, por ejemplo, la Red-RISC-V [2] para la difusión y popularización de un hardware libre basado en el repertorio de instrucciones RISC (Reduced Instruction Set Computer). Esto generará una ventaja importante a empresas y/o universidades, entre otros, para desarrollar nuevas tecnologías basadas en RISC-V sin tener las restricciones de un conjunto muy reducido de fabricantes.

Al ser una arquitectura relativamente reciente aún existen muchos conceptos sobre esta dispuestos a ser explorados lo que nos motiva a introducirla en las prácticas de los alumnos para que puedan conocerla y posteriormente indagar más sobre ella en sus futuros proyectos donde

podrán desarrollar utilizando este tipo de plataformas de bajo coste.

Con este proyecto se pretende introducir RISC-V en la docencia evaluando placas de desarrollo con esta arquitectura para después poder aplicarlas a unas prácticas de laboratorio.

En este informe primero se explicarán algunos conceptos teóricos que nos ayudarán a entender, contextualizar y profundizar mejor el proyecto, seguidamente se comentarán el objetivo general y unos objetivos específicos derivados de este. Después se especificará la metodología que se pretende seguir para poder cumplir estos objetivos y una planificación donde se detallarán los pasos a seguir para el desarrollo del proyecto. Posteriormente se explicará el documento con la introducción al ISA de RISC-V que se ha llevado a cabo y, a continuación, se mostrarán los análisis comparativos y sus resultados. Finalmente, se comentarán los tutoriales del software y las dos prácticas de laboratorio que se han creado y, por último, se profundizará en las futuras líneas de trabajo del proyecto y en las conclusiones finales.

## 2 CONCEPTOS TEÓRICOS

### 2.1 RISC (Reduced Instruction Set Computer)

Conjunto reducido de instrucciones y altamente optimi-

- E-mail de contacte: raul.sierra@e-campus.uab.cat
- Menció realitzada: Enginyeria de Computadors
- Treball tutoritzat per: Dolors Isabel Rexachs del Rosario (departament d'Arquitectura de Computadors)
- Curs 2020/21

zados con un gran número de registros que nos permite obtener un número de ciclos de reloj por instrucción bajo. Las características principales de RISC son la utilización de instrucciones de una sola palabra de tamaño y un acceso a memoria que se ha de llevar a cabo con las instrucciones load y store para poder mover la memoria a registros y así poder realizar las operaciones.

## 2.2 Arquitectura abierta

Son las arquitecturas que se encuentran disponibles gratuitamente y con las cuales se pueden usar, sin necesidad de pagar royalties, para cualquier propósito lo que permite que cualquiera pueda diseñar, fabricar y vender chips y software que utilice esta arquitectura. Este tipo de arquitecturas resultan muy útiles para proyectos de investigación de entidades y organizaciones públicas y/o educativas como por ejemplo en las universidades.

## 2.3 Proyecto EPI (European Processor Initiative)

Es un proyecto co-fundado por la Unión Europea cuyo objetivo es diseñar e implementar una nueva serie de procesadores de bajo consumo europeos para su uso en aplicaciones emergentes y en el Big-Data entre otros. Los procesadores estarán basados en la arquitectura RISC-V y este proyecto forma parte del programa de investigación de la Unión Europea llamado Horizonte 2020 [3].

## 2.4 Red-RISCV

Se trata de una red de investigación sobre las arquitecturas de hardware abierto basado en el conjunto de instrucciones RISC-V formada por centros de investigación y universidades nacionales, entre ellas la Universidad Autónoma de Barcelona. El objetivo principal es crear un ISA reducido, estándar, modular y abierto mediante la colaboración, la evolución conjunta de la investigación, la formación y la innovación que evite o limite los sobrecostos por regalías o desarrollo.

## 3 OBJETIVOS

El objetivo general del proyecto es realizar un estudio sobre RISC-V para posteriormente poder aplicarlo en prácticas y proyectos docentes.

De este objetivo podemos extraer varios objetivos específicos:

- Realizar una búsqueda y síntesis de información sobre la arquitectura RISC-V a partir de la cual se elaborará un marco teórico con conceptos introductorios de RISC-V que se entregará a los estudiantes.
- Analizar placas de desarrollo con la arquitectura RISC-V resaltando sus características y realizar una comparación con placas similares de diferente arquitectura.
- Escoger la placa de desarrollo, el entorno software y los periféricos más adecuados para introducir

la arquitectura RISC-V.

- Diseñar unas prácticas de laboratorio donde se introduzca a los alumnos las características de la arquitectura RISC-V.
- Introducir mediante estas prácticas el repertorio de instrucciones RISC y el funcionamiento de la entrada y salida.

## 4 METODOLOGÍA

A la hora de realizar un proyecto necesitamos predecir el tiempo y los recursos requeridos para completarlo. Cuanto mejor entendamos la metodología que vamos a utilizar tendremos más capacidad de predecir los costes del diseño.

Como metodología del diseño del hardware y las pruebas a realizar se utilizará un modelo de desarrollo en espiral [4], donde mediante un proceso iterativo se irán generando diversas pruebas del sistema, cada una superior a la anterior. En cada fase se utilizará un ciclo de requerimientos/arquitectura/codificación y los resultados de cada ciclo se utilizarán para guiarnos en el siguiente desarrollo. De esta manera, con la experiencia de cada fase nos servirá para ayudar a crear un mejor diseño en la siguiente fase y nos permitirá crear un diseño final óptimo más rápidamente.

Una vez la arquitectura del sistema se ha definido, se empezará a diseñar las prácticas de laboratorio de manera relativa mediante el uso de un flujo de co-diseño. Con este método se pretende conseguir decisiones en la arquitectura apropiadas que permitan la implementación de fases que se pueden llevar a cabo de forma separada. En concreto para este proyecto se irá preparando el material para las prácticas y los documentos complementarios empezando por una primera fase inicial con la estructura general y un seguimiento de fases posteriores con las adiciones y mejoras que se vayan realizando hasta llegar a una fase final con las prácticas y documentos ya preparados para su entrega.

## 5 PLANIFICACIÓN

El proyecto está estructurado en tres partes principales.

### 5.1 Primera parte

Realizar un análisis sobre la arquitectura RISC-V y adquirir conocimientos sobre qué es, sus características, como funciona, etc utilizando como referencia entre otros la guía de RISC-V "Guía Práctica de RISC-V El Atlas de una Arquitectura Abierta" [5]. Sobre el material teórico utilizado, seleccionar las partes más útiles para la elaboración de un documento teórico que se entregará a los estudiantes y nos servirá para ayudarnos en el posterior análisis de las placas con arquitectura RISC-V disponibles.

### 5.2 Segunda parte

Analizar específicamente las placas de desarrollo con arquitectura RISC-V (SparkFun RED-V [6] y SiFive CS-

HiFiveb-01 [7]) que disponemos. Conocer sus características, aprender en qué tipos de sistema funcionan de manera más óptima y compararlas con otras opciones disponibles como la placa ARM Cortex M0+ FRDM k125z [8] que se utiliza en las prácticas de la asignatura Microprocesadores y periféricos para los proyectos de los alumnos. Teniendo en cuenta que estas placas la usan los alumnos de prácticas para sistemas encastados en principio poco complejos, la comparación se centrará principalmente en la disponibilidad de documentación para cada placa, simplicidad, precio y disponibilidad en el mercado y en el consumo. Para este análisis se utilizará los datasheet de las placas entre otro material complementario.

También se realizarán pruebas utilizando diferentes programas para analizar cuál es el entorno de desarrollo más adecuado para diseñar esta práctica con RISC-V comparando las características entre ellos. Al encontrarnos en una situación donde los alumnos pueden no disponer de placa en casa se recomendará utilizar el simulador de RISC-V RARs con el soporte de los videos de Juan González Gómez "Arquitectura de Ordenadores RISC-V" [9].

### 5.3 Tercera parte (Caso práctico)

Como caso práctico se diseñará dos prácticas de laboratorio utilizando las placas con arquitectura RISC-V. Como desde la Universidad Autónoma de Barcelona se quiere apostar por la arquitectura RISC-V, estas prácticas estarán diseñadas para la asignatura de tercer curso Microprocesadores y Periféricos, de la mención de Ingeniería de computadoras del grado en Ingeniería Informática de la Universidad Autónoma de Barcelona. En esta asignatura hasta ahora se utilizaba la arquitectura ARM tanto en la teoría y problemas como en la parte práctica y ahora se pretende añadir la arquitectura RISC-V. Estas nuevas prácticas serán una introducción a la arquitectura para los alumnos, ya que no se ha visto anteriormente en el grado. Primero se hará una práctica introduciendo el repertorio de instrucciones RISC y seguidamente se hará otra práctica donde se introducirán de manera sencilla los conceptos de entrada y salida con periféricos para esta arquitectura.

Durante el transcurso de esta práctica, los alumnos también deberán conocer en qué tipos de sistemas funcionan mejor las placas con arquitectura RISC-V, conociendo sus características para posteriormente poder elegir que tipo de placa es la más óptima para su proyecto.

Para esto previamente se diseñarán e implementarán diferentes códigos para probar diversas funcionalidades y periféricos en la placa con RISC-V, para así poder escoger qué objetivos son los necesarios en la práctica para tener una idea básica de la programación con esta arquitectura.

## 6 INTRODUCCIÓN RISC-V

Como elemento complementario a las prácticas del laboratorio se ha diseñado y creado un documento con una pequeña introducción al repertorio de instrucciones de RISC-V. Para realizar este documento se han aplicado

algunos de los conocimientos adquiridos en la primera parte del proyecto donde el objetivo principal era aprender los conceptos teóricos fundamentales de la arquitectura.

Este documento se ha empezado con una introducción teórica sobre RISC-V y se ha puesto en contexto los motivos de su creciente uso. Después ya se han empezado a explicar las instrucciones más elementales del ISA para poder llevar a cabo los primeros programas básicos en código ensamblador. También se ha hecho una explicación de las pseudoinstrucciones más utilizadas que nos sirven para simplificar nuestro código y hacerlo más legible. Para acabar, se han explicado algunos conceptos relacionados con el ensamblador como el uso de etiquetas o de directivas.

Junto a la explicación de todas las instrucciones y pseudo-instrucciones que aparecen en el documento se incluyen una serie de ejemplos sencillos de código para facilitar el entendimiento a los usuarios. Se ha procurado hacer un documento fácil de leer y bien estructurado para facilitar consultas rápidas, ya que la intención es la de ser un elemento de apoyo a las prácticas de laboratorio. Para ello se han utilizado diferentes colores para diferenciar las distintas partes de los códigos de ejemplo, se han marcado las palabras o frases más importantes y se han creado

#### Instrucción sub

La instrucción para **restar registros** tiene la misma sintaxis que la instrucción **add**, para restar un registro origen 1 con otro registro origen 2 y almacenarlo en un registro destino (RD = RO1 - RO2):

*sub RD, RO1, RO2*

#### Algunos ejemplos de uso:

- Almacenar en el registro x3 la resta de los registros x2 y x1:

*sub x3, x2, x1*

- Asignar al registro x7 el valor del registro x6:

*sub x7, x6, x0 #-- x6 = x6 - 0 = x6*

- Decrementar el registro x4 con el valor de x3:

*sub x4, x4, x3 #-- x4 = x4 - x3*

Fig. 1: Instrucción sub en el documento "Introducción a RISC-V"

unas tablas a modo de resumen de cada grupo de instrucciones y pseudoinstrucciones.

En la Figura 1 se puede observar un ejemplo de la manera con la que se ha diseñado este documento con la instrucción sub del repertorio de instrucciones RISC-V.

Por último, se han añadido la bibliografía utilizada para realizar el documento.

## 7 ANÁLISIS COMPARATIVOS

A continuación, se mostrarán los análisis comparativos, junto a sus tablas comparativas, de las diferentes características de los entornos de desarrollo y de las placas para así tener la información estructurada para poder escoger los más adecuados en el diseño de las prácticas. Estos análisis comparativos forman parte del segundo apartado en la planificación del proyecto.

Tabla 1: Comparación de los entornos de desarrollo

Entorno de desarrollo	Libre y Multiplataforma	Experiencia personal	Ejemplos de uso	Interfaz amigable
Freedom Studio	Si	No, pero interfaz muy similar a Eclipse	Muchos, el propio programa viene con videos explicativos y códigos de ejemplo	Similar a Eclipse y con ayuda de videos del propio programa fáciles para empezar
Platform IO	Si	No	Pocos	Interfaz poco accesible
GNU MCU Eclipse	Si	Si, para la realización de una práctica y con Codewarrior	Muchos, plataforma muy usada en programación	Poco complejo con la experiencia de programar anteriormente
Simulador RARs	Si	No	Algunos vídeos bien detallados	Se aprende rápido con la ayuda de los vídeos

### 7.1 Análisis comparativo del software

Para realizar la comparación del Software para las diferentes placas de desarrollo se han escogido programas de uso libre y que sean multiplataforma (Sistemas operativos Windows, Linux y MAC). Se han seleccionado los programas Freedom Studio, Platform IO y GNU MCU Eclipse para probar la programación en las placas físicas, también se ha añadido en la tabla comparativa el simulador de RISC-V RARs para mostrar sus características pero sin la intención de compararlo con el resto de Software, ya que RARs solo se utilizará para aprender a usar código ensamblador en RISC-V.

Tal y como se observa en la Tabla 1 se ha tenido en cuenta para la comparación la experiencia personal con el programa tanto en la instalación como en la utilización, los ejemplos de uso de los cuales dispone y si su interfaz es accesible y fácil de utilizar.

Sobre la experiencia personal solo se ha tenido contacto previo con el software Eclipse para la programación en Java y a través del uso del programa Codewarrior que utiliza la interfaz de Eclipse. Con el resto de los programas no se ha tenido contacto pero destacamos que la interfaz de Freedom Studio está basada en la de Eclipse por lo que nos ha resultado más fácil para empezar a usar que con Platform IO.

En el apartado de Ejemplos de uso tanto Freedom Studio como Eclipse disponen de mucho material para consultar y aprender. Freedom Studio dispone de muchos programas de ejemplo para adquirir conocimientos sobre la programación en placas RISC-V en el propio programa, después Eclipse, al ser un software muy utilizado para la programación, existe mucho material y ejemplos en diferentes plataformas pero no centrados en la programación específica para placas de desarrollo y en concreto para RISC-V donde este material es más limitado. En el caso de Platform IO la disponibilidad de material y ejemplos de uso es mucho menor.

En interfaces amigables las comparaciones son más complicadas de realizar ya que en general son interfaces parecidas. Destacamos que al disponer de más material de

ayuda tanto con Freedom Studio como con Eclipse han resultado ser más fáciles para poder empezar a utilizarlos de manera cómoda.

Tras un análisis exhaustivo de las características de los diferentes programas y habiendo hecho uso de todos ellos se ha decidido utilizar Freedom Studio por su alta disponibilidad de material y programas de ejemplo y por estar enfocado en el Hardware que utiliza arquitectura RISC-V. Además, se utilizará el simulador RARs como programa complementario para poder aprender sobre el repertorio de instrucciones de RISC-V y como opción para los alumnos que no disponen de las placas de desarrollo en casa.

### 7.2 Análisis comparativo de las placas de desarrollo

En la comparación de las placas de desarrollo se han escogido las tres placas las cuales disponemos, en concreto las SiFive cs-HiHiveb-01 y SparkFun RED-V con arquitectura RISC-V y la FRDM k125z con arquitectura ARM. El objetivo de estas comparaciones no es escoger que placa de desarrollo es la mejor en términos generales, sino mostrar las características y diferencias más relevantes para poder determinar los criterios de selección para utilizar una placa en concreto conociendo las características de las tres que disponemos para analizar.

Como se puede observar en la Tabla 2 para determinar las placas de desarrollo más adecuadas para los diferentes proyectos se ha tenido en cuenta el precio que puede animar a los estudiantes a tener su propia placa, la disponibilidad de material y de ejemplos de uso de las placas y las características en el hardware que disponen. En las primeras tablas comparativas también se habían añadido la facilidad de uso y la complejidad de programación de las placas como elementos a evaluar, pero finalmente se han descartado por los motivos que se explicarán más adelante.

Un aspecto clave a la hora de elegir microcontroladores es el precio, esta característica se ha de contextualizar junto con las características que ofrece cada placa. Destacar que estos precios se han consultado de diferentes vendedores y se ha llevado a cabo una media de los valores. No se

Tabla 2: Comparación de las placas de desarrollo

Placa de desarrollo	Precio	Disponibilidad de material	Ejemplos de uso	Características HW
<b>FRDM K125z (ARM Cortex M0+)</b>	~20€	Datasheet, libro Freescale ARM Cortex-M Embedded Programming	Videos explicativos y ejemplos de código	Deslizador táctil, acelerómetro, Un LED RGB, Flash 128KB
<b>SiFive CS-HiFive-01 (RISC-V)</b>	~50€	Datasheet, esquemático, guía rápida. Información sobre el software en la página web de SiFive	Videos de ejemplo de SiFive	ESP WiFi/Bluetooth, LED RGB, 19 pines digitales E/S 9 pines PWM, Flash 32Mb
<b>SparkFun RED-V (RISC-V)</b>	~35€	Datasheet, esquemático, guía rápida.	Algún video explicativo	LED azul, 19 pines digitales E/S 9 pines PWM, Flash 32Mb

han tenido en cuenta los gastos de envío que suelen ser elevados para este tipo de placas ya que provienen principalmente de Asia, por lo que es muy recomendable la compra de varios productos a la vez para minimizar estos costes extra. Además, hemos de tener en cuenta la disponibilidad de las placas que juntamente con los precios se ha consultado el 20 de junio del 2021 en la última actualización del informe. En precio la FRDM k125z con arquitectura ARM es la más asequible con un precio de alrededor de los 20€, le sigue la SparkFun con un precio cercano a los 35€ y la opción más cara es la SiFive con un precio de unos 50€ y una disponibilidad más limitada.

Tanto en la facilidad de uso como en la complejidad para programar las tres placas no presentan grandes diferencias. Estos factores dependen más de la disponibilidad de material con información de uso y, aunque sean arquitecturas diferentes, a la hora de programar tanto ARM como RISC-V funcionan de forma muy parecida.

En la disponibilidad de material las dos placas de RISC-V tienen a su disposición un Datasheet [10], un esquemático [11] [12] y una guía rápida [13] [14], aparte en la página oficial de SiFive, compañía de la placa HiFive, se encuentra información útil como diferentes programas para empezar a usarla. En el caso de la FRDM k125z dispone de datasheet y existe un manual muy completo sobre programación de sistemas embebidos para ARM Cortex-M [15].

Los microcontroladores FRDM y SiFive disponen de varios ejemplos de uso tanto en formato escrito como en vídeos, en cambio, la SparkFun al ser una placa bastante nueva prácticamente no tiene ejemplos, aunque funciona de una forma similar a la SiFive.

Las características hardware que ofrece cada placa nos determina en gran medida la elección para un proyecto determinado. El RED-V y la SiFive HiFive-01 son prácticamente iguales en hardware. Las únicas diferencias son

que la placa SiFive tiene un dispositivo ESP WiFi/Bluetooth y también tiene un LED RGB a diferencia de la SparkFun con solo un LED azul y ningún dispositivo ESP. En los dos casos disponen de 19 pines digitales de entrada y salida, 9 pines PWM y una Memoria Flash de 32Mb. La frdm k125z tiene la ventaja de disponer aparte de un LED RGB un acelerómetro y un control deslizante táctil como periféricos. En memoria Flash se queda algo más corta que las dos placas RISC-V con un tamaño de 128KB.

Finalmente, no se han podido hacer pruebas para medir el consumo por falta de tiempo y recursos, esto se comentará como futura línea de trabajo más adelante en el informe.

Concluyendo y aunque no exista una notable diferencia entre las tres placas, la FRDM k125z se podría considerar la mejor elección para sistemas embebidos pocos complejos gracias a su asequible precio, mayor variedad en periféricos y gran cantidad de ejemplos de uso que dispone. En el caso de querer utilizar una arquitectura RISC las dos opciones analizadas son muy parecidas en muchos aspectos. Quizás el precio puede hacernos decantar por la SparkFun aunque al ser una placa de desarrollo bastante nueva disponemos de pocos ejemplos de uso todavía.

## 8 TUTORIALES SW

Después de realizar el análisis comparativo del software y haber escogido los programas que se iban a utilizar para la realización de las prácticas, se empezó a diseñar y ejecutar los documentos complementarios a las prácticas con los tutoriales de estos programas.

### 8.1 Tutorial Freedom Studio

Tras el análisis comparativo entre los diferentes entornos de desarrollo Freedom Studio fue el escogido para ser utilizado para la programación de las placas. En la Figura 2 podemos observar la página principal de este programa.

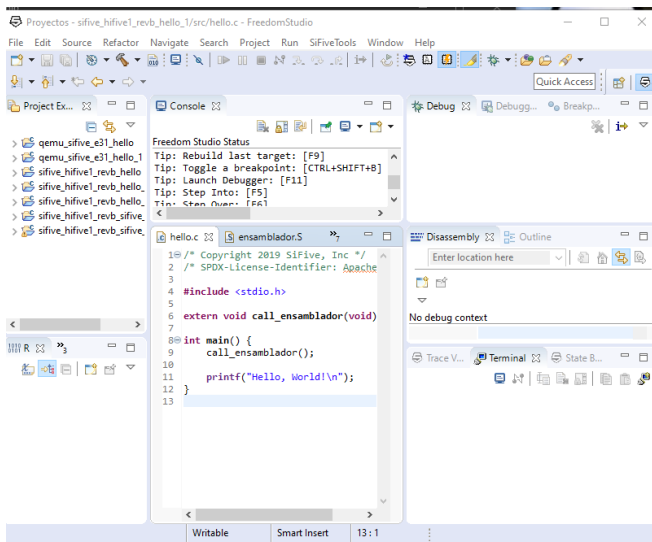


Fig. 2 Página principal del programa Freedom Studio

En el tutorial del programa Freedom Studio para la programación de las placas de desarrollo RISC-V primero de todo se ha hecho una introducción al programa explicando sus características más importantes. A continuación, se ha explicado como llevar a cabo la instalación del programa para Windows, Linux o Mac haciendo hincapié en las particularidades que presentan cada sistema operativo para no tener problemas más adelante a la hora de ejecutar este software.

Una vez hecha la instalación se han presentado los pasos para crear un primer programa en la placa de RIISC-V utilizando uno de los muchos programas de ejemplo que contiene Freedom Studio. Tras ejecutar nuestro primer código y habiendo visto las opciones que nos ofrece como la opción de Debugar, se ha mostrado como hacer un primer programa simple utilizando instrucciones de ensamblador RISC-V. Para ello se ha escogido uno de los programas de ejemplo que se encuentran escritos en C y desde el código se ha realizado una llamada a un archivo ensamblador donde hemos puesto las instrucciones máquina. En este apartado también se ha mostrado donde ver los valores de los registros al ejecutar el código paso a paso en el modo Debug y así poder entender mejor la ejecución del programa realizado.

Para acabar, se ha presentado la bibliografía utilizada para llevar a cabo este tutorial haciendo énfasis en la recomendación de echar un vistazo al completo manual de Freedom Studio que se encuentra en la página oficial en el caso de necesitar más información sobre el funcionamiento del programa.

## 8.2 Tutorial Simulador RARs

Como se ha expuesto anteriormente, utilizaremos el simulador RARs para los alumnos que no disponen de placa o que necesitan hacer pruebas, y así tengan la posibilidad de ejecutar código ensamblador de RISC-V. Por este motivo se ha diseñado y creado un documento con

un tutorial sobre este simulador. En la Figura 3 se muestra la página principal del programa.

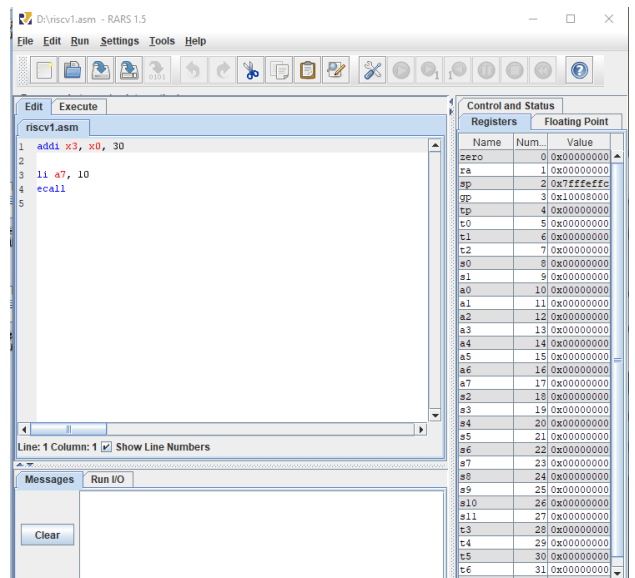


Fig. 3 Página principal del simulador RARs

Aunque el programa contiene bastantes opciones, en el documento se ha hablado de lo más fundamental empezando con una pequeña explicación teórica sobre el simulador. Seguidamente se han presentado los pasos para realizar la instalación en Linux i en Windows y se ha mostrado como realizar un primer programa en código ensamblador RISC-V. También al ser un simulador, se ha procurado exponer como funciona el modo Debug, donde se pueden ver los valores de los registros en todo momento y como se puede ejecutar el programa con un número de instrucciones ejecutadas por segundo limitado manualmente.

En la parte final del documento se ha indicado la página de donde se ha extraído la información para hacer el tutorial en la que se encuentran más opciones del simulador RARs explicadas con mayor profundidad.

## 9 PRÁCTICAS DE LABORATORIO

En este apartado se hará una explicación del contenido de las prácticas de laboratorio realizadas para este proyecto con la finalidad de introducir la arquitectura libre RISC-V a los alumnos.

Se han llevado a cabo dos prácticas, una introductoria tanto a RISC-V como a las placas de desarrollo y software y una segunda centrada en la entrada y salida.

Ambas prácticas contienen unos requisitos previos antes de empezar a realizar la práctica para que los alumnos tengan una idea general de lo que luego se va a tratar en la práctica. También, tienen unos objetivos que determinarán lo que se quiere conseguir al finalizar la práctica y estos puntos nos servirán al final para comprobar su cumplimiento y tener unos criterios de evaluación. Por

último, en las dos prácticas se presentan cuales son los entregables que se van a hacer al final junto a la forma de entrega y una bibliografía con las fuentes consultadas en la creación de la práctica y que pueden ser de gran ayuda a los alumnos para su realización.

Estas prácticas contienen aparte un documento pensado para los docentes con información sobre como llevarlas a cabo, como evaluarlas, unas soluciones orientativas de los ejercicios y preguntas propuestos a lo largo de la práctica y ejercicios extra para proponer a los alumnos que hayan acabado las prácticas antes del tiempo estimado de la sesión.

### 9.1 Primera práctica

La primera práctica se ha dedicado a la introducción tanto de RISC-V como a las placas de desarrollo y los entornos software que se van a utilizar.

Primero de todo, en el documento se ha hecho una introducción teórica sobre RISC-V resaltando los elementos más significativos que la definen y aportando un contexto a el uso de esta arquitectura en las prácticas de la asignatura.

Más adelante se ha hablado sobre las dos placas de desarrollo que se pretenden utilizar en las prácticas. Se han destacado las características y las partes que las conforman a cada una mediante el uso de una imagen de modo que se puedan ver de manera más clara los detalles.

Seguidamente, se ha empezado a comentar los entornos de desarrollo que se van a utilizar en las prácticas, mencionado los dos tutoriales para su instalación y uso. En la propia práctica se han cogido partes del documento de tutorial de Freedom Studio realizado anteriormente para que los alumnos realicen un primer programa en RISC-V con él y así familiarizarse. También se ha comentado la posibilidad de utilizar el simulador RARS o incluso el propio simulador del que dispone Freedom Studio para la realización de esta práctica.

Para acabar, se han diseñado 5 ejercicios con los que comprobar el correcto desempeño de lo presentado en la práctica y concretamente en los objetivos de esta. En los dos primeros ejercicios se da un código ya escrito para ejecutar y depurar para después contestar unas preguntas teóricas. Finalmente, en los tres últimos ejercicios se propone la creación de un programa y seguidamente contestar unas pocas preguntas sobre este. En el documento para los profesores se han incluido dos ejercicios extra para proponer a los alumnos que acaben la primera práctica antes de la hora prevista de la sesión en el laboratorio

### 9.2 Segunda práctica

En la segunda práctica de laboratorio creada se ha hecho una introducción a la entrada y salida procurando profundizar más en el repertorio de instrucciones de RISC-V.

Para empezar, en el documento se ha señalado los moti-

vos por los cuales es importante la entrada y salida en este tipo de sistemas para poder darle un contexto a esta práctica.

Posteriormente se han realizado una serie de explicaciones teóricas para entender mejor como funciona el concepto de entrada y salida y se ha mostrado una manera de encender un conjunto de LEDs con código para comprender mejor como llevarlo todo a la práctica.

También, se ha aprovechado la función que contiene el programa RARS para hacer uso de la simulación de dos visualizadores de 7 segmentos con los que poder experimentar. Se ha llevado a cabo una explicación detallada y un ejercicio práctico sencillo para comprobar su correcta comprensión.

Por último, se ha diseñado un mini proyecto a elección de cada grupo de dos o tres alumnos donde han de crear un programa con la funcionalidad que se prefiera pero que utilice LEDs o cualquier otro tipo de dispositivo de salida y también utilice algún dispositivo de entrada. Este programa ha de ser sencillo ya que está pensado como entrega de esta práctica, pero se valorará bastante que tenga una funcionalidad creativa. Con este ejercicio final se quiere hacer hincapié en el resultado de aprendizaje definido en la guía docente de la asignatura de desarrollo de la curiosidad y la creatividad, al diseñar programas por parte de los alumnos con los recursos disponibles en el laboratorio. Además, se remarca en el documento que se valorará muy positivamente que en el entregable final de la segunda práctica se realice una presentación del proyecto de forma escrita clara y concisa que nos preparará para las futuras presentaciones de los proyectos de la asignatura.

Como alternativa, a los alumnos que no dispongan de placa de desarrollo se les propondrá hacer el mismo proyecto utilizando el simulador RARS. En estos casos, al no disponer de periféricos para interactuar con el programa se utilizará la simulación de los dos visualizadores de 7 segmentos que contiene el programa como dispositivos de salida y se utilizará un dispositivo de entrada ficticio a elección para el programa creado.

## 10 FUTURAS LÍNEAS DE TRABAJO

Aunque en este proyecto se han podido cumplir en gran medida los objetivos iniciales propuestos aún existen futuras líneas de trabajo y propuestas de mejora

Existen aún muchos conceptos sobre RISC-V por explorar y profundizar ya que para llevar a cabo este proyecto solo se ha investigado la parte más fundamental y esencial de la arquitectura para introducirla en la docencia.

También existen muchas más placas de desarrollo RISC-V que se pueden analizar y proponer para su uso en la docencia, sin tener en cuenta los nuevos microcontroladores que irán saliendo a lo largo de los años. Por lo tanto, una



revisión de las placas cada año puede ser una buena manera para ir actualizando estas prácticas. Además, existe la posibilidad en un futuro de crear un propio microcontrolador RISC-V con las características necesarias para la docencia al ser hardware libre y modular.

En el análisis comparativo de las placas de desarrollo se pretendía realizar pruebas en el consumo de cada una ya que se consideraba un aspecto a tener muy en cuenta en la comparación de microcontroladores. Estas pruebas no se acabaron de hacer por falta de recursos y mala organización del tiempo. En un futuro sería una buena opción hacerlas para disponer de un criterio extra en la comparación de las placas.

Durante la creación de las prácticas de laboratorio se planteó la posibilidad de realizar pruebas en alumnos que se encontraban cursando el grado. Finalmente, puesto que las prácticas eran de larga duración y hubo falta de tiempo estas pruebas no se llevaron a cabo. Al no haberse probado aún con los alumnos, las prácticas realizadas se encuentran en un proceso de mejora continua apoyada por las opiniones tanto del alumnado como de los profesores que las imparten.

Una de las ideas que surgieron mientras se hacían los diferentes tutoriales fue, aparte de los documentos, crear una serie de videos cortos explicando muchos de los conceptos por el hecho de que, en general, los alumnos están más acostumbrados y les resulta más fácil y ameno aprender utilizando medios visuales que escritos. Esta idea se pensó llevar a cabo si se conseguían acabar todos los tutoriales y prácticas con tiempo, ya que estos videos requerían de una preparación previa. Por desgracia, esos videos finalmente no se han podido realizar y quedan como propuesta de mejora para las futuras prácticas de laboratorio.

## 11 CONCLUSIONES

En este apartado se van a presentar unas conclusiones basadas en el cumplimiento de los objetivos definidos al inicio del trabajo.

Como elemento complementario para la contextualización de las prácticas de laboratorio y como resultado de la parte 1 de la planificación del proyecto, se ha creado un documento de introducción al repertorio de instrucciones de RISC-V donde se han aplicado los conocimientos adquiridos sobre esta arquitectura. Este documento me ha ayudado a conseguir sintetizar los conceptos más fundamentales y, de esta manera, comprender mejor el funcionamiento del ISA de RISC-V.

Con la ayuda de las comparaciones expuestas en este documento de los entornos de desarrollo y de las placas y con las pruebas realizadas de diversos códigos se ha logrado seleccionar el software más adecuado para poder introducir la arquitectura RISC-V a los alumnos y conocer

mejor las placas de desarrollo que se van a utilizar.

También, se ha obtenido una alternativa en el desarrollo de código RISC-V para los alumnos que no puedan disponer de una placa con la ayuda del simulador RARS introduciendo, mediante un documento, un tutorial de este programa.

Además, se han conseguido diseñar y crear dos prácticas de laboratorio con las que los alumnos podrán aprender y experimentar sobre RISC-V con microcontroladores de hardware abierto. Con estas prácticas se han materializado las diferentes pruebas realizadas con el código y los periféricos durante los análisis comparativos, seleccionando las más eficaces para comprender mejor los conceptos marcados como objetivos en las prácticas.

Para acabar, como se ha comentado anteriormente, aún existen futuras líneas de trabajo y mejoras para este proyecto, pero considero personalmente que he conseguido llevar a cabo una buena base para poder incluir progresivamente el hardware libre en la educación y de esta manera fomentar su continua investigación y desarrollo por parte de las universidades y otros tipos de entidades colaboradoras.

## AGRADECIMIENTOS

Primero de todo me gustaría agradecer a Diego Mostaccio por proponer este trabajo y darme la motivación para llevarlo a cabo, a Dolores Rexachs por hacerse cargo de mi tutorización y darme soporte con los problemas que me surgían en la realización del trabajo y a Daniel Ruíz por interesarse en él y ayudarme con la impresión en 3D de las cajas para las placas de desarrollo. También me gustaría agradecer a aquellos docentes que durante el transcurso de la carrera de Ingeniería Informática me han aportado los conocimientos y la motivación necesaria para querer continuar con el grado a pesar de mis dudas iniciales. Finalmente me gustaría agradecer a todas las personas que durante estos últimos años han confiado en mí, me han apoyado y han estado presentes tanto en los buenos como en los malos momentos. Puedo asegurar con la cabeza bien alta que todo el amor que he recibido estos años me ha servido muchísimo para poder seguir por el buen camino.

## BIBLIOGRAFIA

- [1] "EPI." european-processor-initiative.eu. <https://www.european-processor-initiative.eu/project/epi/> (accedido Mar. 11, 2021)
- [2] "Red-RISC-V" uab.cat <https://www.uab.cat/web/sala-de-premsa/detall-de-noticia/es-crea-la-xarxa-riscv-per-impulsar-el-desenvolupament-del-hardware-de-codi-obert-1345667174054.html?noticiaid=1345806981489> (accedido Jun. 1, 2021)
- [3] "Horizon 2020." ec.europa.eu. <https://ec.europa.eu/programmes/horizon2020/en> (accedido Mar. 11, 2021)



- [4] Marilyn Wolf, "Design methodologies," in High-Performance embedded computing: Applications in Cyber-Physical Systems and Mobile Computing, Georgia Institute of Technology: Morgan Kaufmann, 2014, cap. 1, pp. 20-31.
- [5] David Patterson y Andrew Waterman, Guía Práctica de RISC-V El Atlas de una Arquitectura Abierta. San Francisco, California, USA: Strawberry Canyon LLC, 2018.
- [6] SparkFun Dev-15594, Sparkfun Electronics, Disponible: <https://www.mouser.es/ProductDetail/SparkFun/DEV-15594?qs=sGAEpiMZZMu3sxp5v1qrhi\%2FDDEKTIJRJ\%252BNvNIRYSlt=>
- [7] CS-hifiveb-01, Crowd Supply, Disponible: <https://www.mouser.es/ProductDetail/Crowd-Supply/cs-hifiveb-01?qs=\%2Fha2pyFaduiFwrVXAYQ1Di6\%2FIfpIpOJW8Com8Cr8aKE\%3D>
- [8] ARM Cortex-M0+, ARM Developer, Disponible: <https://developer.arm.com/ip-products/processors/cortex-m/cortex-m0-plus>
- [9] Juan Gonzalez Gomez. Arquitectura de Ordenadores RISC-V. Práctica 1-Sesion-1-Video-01 (Set. 29, 2019) Accedido: Feb, 14, 2021. [Video Online]. Disponible: <https://www.youtube.com/watch?v=GYvBAHdkRwk>.
- [10] SiFive, Inc. SiFive FE310-G000 Manual v2.3. (2017). Accedido: Abril 17, 2021 [Online]. Disponible: <https://www.mouser.es/pdfDocs/FE310-G000.pdf>
- [11] SiFive, Inc. HiFive1 Rev B Schematics (2021). Accedido: Abril 17, 2021. [Online]. Disponible: [https://sifive.cdn.prismic.io/sifive/c34f4c7f-0d3a-493e-8a19-a0b18f8a4555\\_hifive1-b01-schematics.pdf](https://sifive.cdn.prismic.io/sifive/c34f4c7f-0d3a-493e-8a19-a0b18f8a4555_hifive1-b01-schematics.pdf)
- [12] SparkFun, RED-V RedBoard Schematics. Accedido: Abril 17, 2021 [Online]. Disponible: [https://www.mouser.es/pdfDocs/SparkFun\\_RED-V\\_RedBoard-Schematic.pdf](https://www.mouser.es/pdfDocs/SparkFun_RED-V_RedBoard-Schematic.pdf)
- [13] SiFive, Inc. SiFive HiFive1 Rev B Getting Started Guide v1.2. (2021). Accedido: Abril 17, 2021. [Online]. Disponible: [https://sifive.cdn.prismic.io/sifive/cf239fd0-ae4f-4fd8-a944-fdafb5018153\\_hifive1b-getting-started-guide\\_v1.2.pdf](https://sifive.cdn.prismic.io/sifive/cf239fd0-ae4f-4fd8-a944-fdafb5018153_hifive1b-getting-started-guide_v1.2.pdf)
- [14] SparkFun, RED-V RedBoard Hookup Guide (2019). Accedido: Abril 17, 2021 [Online]. Disponible: <https://www.mouser.es/pdfDocs/RED-VRedBoardHookupGuide-learnsarkfuncom.pdf>
- [15] Sarmad Naimi, Sepehr Naimi y Shujen Chen, Freescale ARM Cortex-M Embedded Programming. Freescale, 2016.

## APÉNDICE

### A1. Imágenes del material utilizado durante el proyecto

#### Placas de desarrollo:



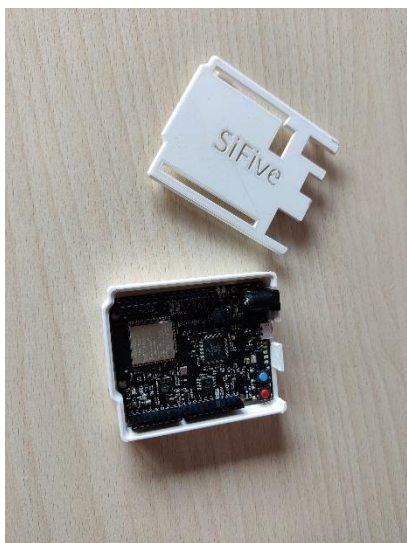
*FRDM k125z (ARM)*



*SparkFun RED-V (RISC-V)*



*SiFive CS HiFiveb-01 (RISC-V)*

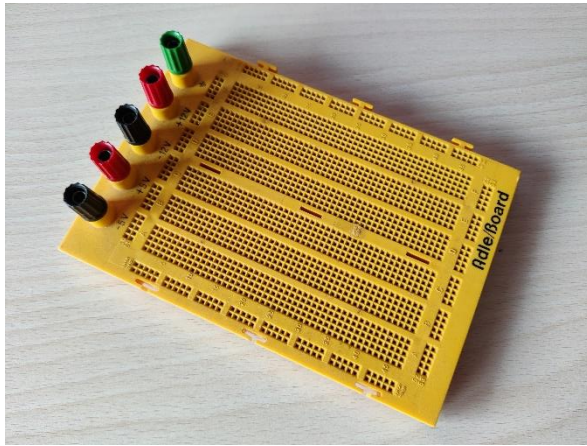


*SiFive CS HiFiveb-01 con caja Impresión 3D*

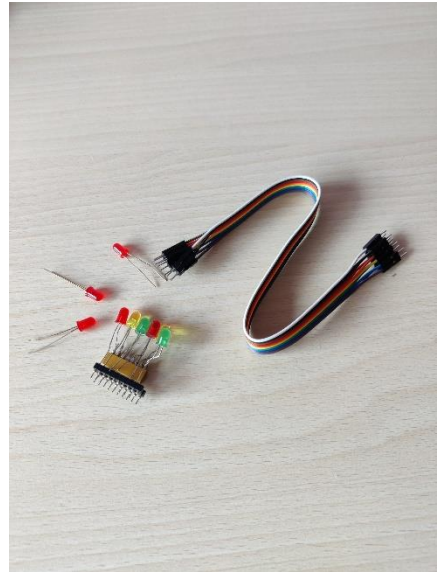


*SiFive CS HiFiveb-01 con caja Impresión 3D*

## Material complementario:



*Protoboard*



*Pines macho-macho y LEDs*